Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 23 (2011) 65 – 71

Procedia Engineering

www.elsevier.com/locate/procedia

JSF-based Framework for Device Management System Design and Research

WANG Meng[1] ,WANG Yan-en[2] ,ZENG Wen-Xiao[1] ,Yao Chen[3]

[1.] *The Xinjiang Technical Institute of Physics & Chemistry. CAS*
[2.] *Dept. School of Mechatronics Northwestern Polytechnical University Xi'an,China*
[3].*Department of Computer Science,Shangluo University Shangluo, China*

Abstract

With the rapid development of global integration ， the mode of equipment management is changing to the trend of network and automation. Against this trend , the article analyses advantage of B/S structure, compares the performance and characteristic of several J2EE application frameworks, considers users' future development needs and system expansion based on the internet equipment management system of web forms, the workflow technology is drawn into to optimize the process of equipment management and increase the extent of automation.

Keywords : workflow; workflow engine; JSF framework; OSworkflow

1. Main text

Technology to use Hibernate for persistence operations. Persistence layer encapsulates the business data access details for the business layer provides an object-oriented API. It does all the data access functions, the upper incoming data is written to persistent storage system, and in accordance with the requirements of the upper read or modify existing data.

1.1. Persistence domain model design:

Persistence layer using Hibernate as the ORM technology, so you can fully object-oriented approach to the database. Ensure that the entire process of object-oriented systems development approach to analysis, design and programming. Hibemate Although the use of the HQL to mask the differences between databases, but it is still compatible with SQL, and SQL did not block out the power in the persistence layer and database tables corresponding to the first write data class, write the corresponding Hibemate

doi:10.1016/j.proeng.2011.11.2466

mapping configuration file * . hbm.xml, write DAO interface and corresponding implementation class. Hibernate provides data recovery and updating tools, data source transaction management, connection pooling and other management data.

⑴Design domain model entities

Application of object-oriented design approach, the use of advanced design entity domain model, to determine their relationship, then the DDL provided by the Hibernate tools to automatically map the physical domain model for the database table structure. The system is divided into six modules designed a total of 21 entity types. Waste management module in which three entities were designed class. The retirement system module entity class model diagram.

⑵ Create entity domain model class

In the system development process, using a Hibernate3.2 version and Java5 Annotation support for the domain object mapping to the database table structure with all the Hibernate Annotation written in class, so traditional counterparts, each domain object Hibernate mapping file (. hbm.xml) no longer needed. This makes the code easier to maintain.

⑶ Map domain model entity classes

As a result of the Hibernate Annotation, on the steps and step into one .

1.2. DAO component design of persistence

⑴ DAO interface definition

DAO DAO interface is the component that contains various processing methods provide a statement. DAO component approach is not designed outset in the design process, first determine the components of this system, a common method of DAO. These methods are:

Save: save the persistent objects.

Delete: delete persistentλ objects

Update: update the persistent object

Get: access to persistentλ objects based on OID

getAll: get a list of allλ persistent instances

contractkeQuisitionDAO interface has basic database operation method: one method to apply the query get, one way to save the application save, one way to apply updates Update, delete the application form method delete, and finddBy deviceId, findAll method. Application uses two DAO interface - CatalogDao and UserDao for hibernate related data management and persistence logic.

⑵ DAO interface class

Implementation of DAO interface class on ly.dao.hibernate package. Spring HibernateTemplate class used to support the operation of the Hibernate persistence, you can achieve the integrity of the Hibernate package. Can be very time-saving to complete the implementation of DAO interface. The use of such methods can simplify Hibemate getHibernateTemplete database access operations. Thus easier to query the database, add, modify, delete, do not focus on the database connection management or exception handling these things.

⑶ Container instead of DAO factory with SpringIOC

Through Spring's IOC can not modify any code in the case for libraries to achieve a seamless migration of the database. Spring IOC container according to the demands of specific DAO instance. The DAO components configured in the Spring configuration file in the specified container, then use it to manage the DAO components ApplicationContext lifecycle. This used to take the initiative by the business layer to get the case being transferred to DAO DAO is set to take the initiative to the Business Logic Layer. IOC makes this loose coupling between modules, the application is more convenient.

1.3. the data layer to generate the table structure

org.hibernate.tool.hbm2ddl.SchemaExport is Hibernate3, 2 a useful tool in the class, it can be done to the database into entity relational model domain model of automatic conversion, database table structure to obtain the DDL scripts, you need to do preparatory work is to modify the Hibernate configuration file, you need to do the following two steps.

write Hibernate.cfg.xml profile

Although the Spring Framework does not have to write the configuration file can directly generate the Hibernate SessionFactory, however, SchemaExport class need to create the file SessionFactory. Therefore, the project's classes directory in a good place the hibernate.cfg.xml file.

the preparation method to generate a database script calls SchemaExport class

Write a method in the method call SchemaExport class to generate a database script, or directly generate the database table structure (you must first start the database service, and created the database), then the main method call this method,

Run this class will be in the current project directory DBScript directory called jsh-mysql.sql generated script file. You can then use this script to create the table structure of the database. Create the table structure before SQLserver created in the database encoding to utf-8. Also in the current project directory DBScript directory has a data.sql script, is the application of some sample data. In order to demonstrate the correct operation of this project, run the program before the first script in the "jsh_petstore" database to perform again.

1.4. Business Logic Development

Business objects and business services in the business logic, they combine to provide a more advanced business logic. Can be defined using the business interface, the interface should be standard, common. The system service layer by the spring of the IOC to complete.

1） Business logic interface design: the business logic from the business logic components of the DAO components Façade package, reducing the business layer and presentation layer of the coupling, the separation of interface and implementation, business logic layer provides the interface in the business layer that defines the function at the same time to achieve specific business logic written in its corresponding implementation class; such as business services on the user interface, it defined the methods include user login and search, etc., can be called WEB layer. This advantage is when the user interface changes when the business will not affect the web layer. Similarly, the logic layer and the coupling between the DAO is true, just call the DAO layer of the component interface. This enables the logic layer and the loose coupling between the DAO.

2） Exception handling

Error when the service request is very much an error exception will be generated so as to disrupt the process, and should report the abnormal level up, making the process can follow a predefined procedure successfully implemented. Therefore, in defining business logic interface, each method should be declared to the custom business to throw exceptions. Custom exception class code is as follows:

```
package org.qiujy.model.exception;
/ ** Custom business exception * /
public class MyException extends Exception {
public MyException () {
super ();
}
public MyException (String arg0, Throwable arg1) {
```

```
super (arg0, arg1);
}
public MyException (String arg0) {
super (arg0);
 }
public MyException (Throwable arg1) {
super (arg1);
 }
}
```

3 ）Business logic components

Business logic components in the component by calling the DAO persistence operations to complete the appropriate business logic processing, and must therefore be accepted into the Spring container DAO components, and must implement the business logic components to provide the corresponding setter methods for class . In the implementation class, the main components of the total call DAO methods to implement business logic processing, while the process to record the error message, use the apache commons-logging.jar log processing package.

4） Assembly of business logic components

The design of the DAO components, DAO component initialization by the spring dependency injection mechanism to complete. Business logic components for the assembly industry by the spring to do, so the business logic component configured in the configuration file applicationcontext.xml in the spring. The application of the six required business logic to manage the class configuration to the spring.

1.5. JSF and Spring integration

Spring middle tier with the JSF Web-tier consolidation easiest way is to use Spring's org.springframeword.web.jsf.DelegatingVariableResolver.. DelegatingVareableResolver first query assigned to the JSF's default parser. And is Spring's WebApplicationContext. JSF managed Bean so you can use Spring injection to manage. Business needs to display a Spring-managed Bean for instance, you can use this tool org.springframeword.web.jsf.FacesContextUtils class. It is based on the current FacesContext to get a Spring ApplicationContext instance, from the ApplicationContext instance can easily get the desired instance of a Spring-managed Bean.

1.6. Web-layer design

WEB-layer design idea: the page the user visits the page some initial data from the JSF component "pull out", while triggering event page by clicking on the button to complete the action of a trigger, the request has been sent to the JSF Managed Bean , then the corresponding Managed Bean method is called the corresponding event handling business logic components to the user request processing, and finally navigate to the results page displays the results.

Managed Bean Design: Design a Managed Bean class ---- baseBean base class to handle all managed access to business logic components Bean problem. All Managed Bean abstract base class is mainly responsible for loading serviceLocator instance, in addition to providing a init () method to initialize the subclass attributes. BaekingBean, APplieationBean are inherited from baseBean. BackingBean role is to define a page in the Ul component and the associated property, and the other is to define a logical way to deal with the web tier. BackingBean deal with web-tier controller logic is an extension.

Page layout program: page layout using a relatively easy way to understand include, into the page header section, footer section, part of the menu bar, main content section. Regarded each page header

section, footer section, include some of the menu bar to come, but only need to redefine the main content section. Each page needs to import the stylesheet and javascript file, so write it to separate meta.jsp file, on each page using the include directive to import. Set in the header part of the main site logo, and the search code, the page tab at the top level component use <f:subview>. Used to display some footer file hyperlinks. Menu bar is part of the main functions of the current application in all major categories of equipment listed, when initializing applicationBean get from the database necessary for all major categories of equipment, so have concerns applicationBean configuration and implementation.

2. OSWorkflow in the system to achieve

This section will be scrapped equipment, workflow, for example, the module table with scrap the design process steps to achieve through this module OSWorkflow to detail the configuration process and application.

2.1. Required JAR files by OSWorkflow

As shown in Table 1 jar files required for the system

Table 1 jar files

Classification	Package name required	If necessary
OSWorkflow Need to package their own	osworkflow-2.8.0.jar	Yes
OSWorkflow Packet core references	commons-logging.jar	Yes
	propertyset-1.4.jar	Yes
	oscore-2.2.5.jar	Yes
OSWorkflow Optional package	bsf.jar	No
	ehcache.jar	No
	osuser-1.0-dev-2Feb05.jar	No
	bsh-1.2b7.jar	No
Combined with the required packages with Spring2	spring.jar(version:2.05)	Yes
Combined with the required packages with Hibernate3	antlr.jar	Yes
	log4j.jar	Yes
	dom4j.jar	Yes
	jta.jar	Yes
	hibernate3.jar	Yes
	commons-collections.jar	Yes
	asm-attrs.jar	Yes
	asm.jar	Yes
	cglib.jar	Yes

Package name is required for the necessary classification
OSWorkflow own needs necessary package osworkflow-2.8.0.jar
OSWorkflow packet core references need commons-logging.jar
propertyset-1.4.jar necessary
oscore-2.2.5.jar necessary
OSWorkflow optional optional package bsf.jar
ehcache.jar Optional
osuser-1.0-dev-2Feb05.jar Optional
bsh-1.2b7.jar Optional
And Spring2 combined with the required package spring.jar is necessary
Combined with the required packages with Hibernate3
antlr.jar necessary
log4j.jar necessary
dom4j.jar necessary
jta.jar necessary
hibernate3.jar necessary
commons-collections.jar necessary
asm-attrs.jar necessary
asm.jar necessary
cglib.jar necessary

2.2. Whole process of concept

OSWorkflow engine is only responsible for the "process operation", the operation will be based on your definition of Action and condtion of the judge. Function of OSworkflow, is a step, action, result of the implementation process need to call the function, as this function for what, OSWorkflow do not care about the engine is only responsible for providing several parameters interface. Process steps defined in the 0S__FLOWSTEP table, running the system in the process of execution of the process to determine the next steps of the process steps based on the table Id. The following table 2 is based the equipment scrapped process information.

Table 2 Process equipment design table

Id	Name	Forward	RoldID	FlowType
10021	Retirement application form submitted	10022	2327	equipPutFlow
10022	Division Head approval of motor	10023	2347	equipPutFlow
10023	The development of retirement programs	10024	2327	equipPutFlow
10024	Division Head approval of motor	10025	2347	equipPutFlow
10025	Submit a single waste storage	10026	2327	equipPutFlow
10026	Archive	19999	2343	equipPutFlow

Id Name Forward RoldID FlowType
10021 to submit application form 10022 2327 equipPutFlow scrapped
Mobile 10022 10023 2347 equipPutFlow approval of the Officer
10023 10024 2327 equipPutFlow develop retirement programs
Mobile 10024 10025 2347 equipPutFlow approval of the Officer
Submit 10025 10026 2327 equipPutFlow single waste storage
10026 archive 19999 2343 equipPutFlow
Field Description: ID and name represent the step number and name; Forward: the next step specified number; RoldID indicate which role to perform this step; FlowType shows the type of process.

3. Conclusions

And processes to achieve focus from functional to achieve the realization of two aspects about the system. Functions to achieve important aspects to describe the three levels, namely the implementation of persistence layer, business logic implementation, and web tier implementation. Realization process to equipment obsolescence management process as an example of the configuration process step and a brief overview and description.

Acknowledgment

The work was supportted by the Department of Science & Technology of Xinjiang Uygur Autonomous Region (201191115)

References

[1]BruceEchel.Thinking in Java 2nd Edition[M].President Mind View Inc，1998
[2]Bruce Eckel.Thingking in Java Fourth Edition[M].Manning Publications，2004
[3]Rod Johnson.Expert One-on-One J2EE Design and Development[J].Wrox，October 23，2002
[4]Hofmeister.C，NordR.L，SoniD.Applied Software Architecture[M].Addison Wesley，2005
[5]Manuel D.Rossetti， Bradley J.Hobbs， Paul D.Faas.An Object-Oriented Framework for Simulating Automatic Data Collection Systems[M].Faas，Dec，2006
[6]Jim Coker.Object Persistence and Distribution.http://developer.java.sun.com
[7]Tyler Sperry.Solving the Java Component Puzzle.http://www.java-pro.com
[8]Robert.Mastering Enterprise Java Beans and the Java 2 Platform.Enterprise Edition.Elliott[M].John Wiley&Sons，1999
[9]Hans.Bergsten.Java Server Faees[M],ReillyMedia,Inc,2004
[10]Spring framework. org. Spring Framework Documentation [EB/OL]. 2005.